PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : H04L 29/06, 9/08

A1

(11) International Publication Number: WO 93/09627

(43) International Publication Date: 13 May 1993 (13.05.93)

(21) International Application Number: PCT/CA92/00486

(22) International Filing Date: 9 November 1992 (09.11.92)

(30) Priority data:
07/789,275 8 November 1991 (08.11.91) US

(71)(72) Applicants and Inventors: LEE, Ernest, Stewart [CA/CA]; 15 Westridge Road, Etobicoke, Ontario M9A 4E6 (CA). THOMPSON, Philip, Martin [CA/CA]; R.R. #3, Perth, Ontario K7H 3C5 (CA).

(74) Agent: ORANGE, John, R., S.; Sim & McBurney, 330 University Avenue, Suite 701, Toronto, Ontario M5G 1R7 (CA).

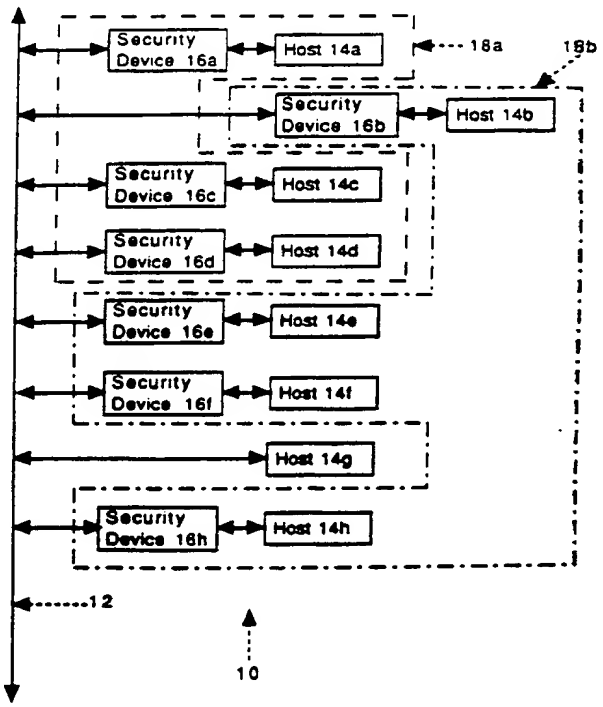
(81) Designated States: AU, CA, JP, KR, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: CRYPTOGRAPHIC APPARATUS AND METHOD FOR A DATA COMMUNICATION NETWORK

(57) Abstract

A data transmission network (12) has a plurality of computers (14) interconnected by a transmission channel (12). The computer communicates with the channel through a security device (16) which encrypts and decrypts data. The device uses a key packet distributed over the network from which a new key is derived by using the encryption process within the device. The encryption process makes use of a data sequence or "secret" peculiar to each domain so that the key generated in the domain is also peculiar to that domain. The key is changed as the encryption proceeds and upon completion of the data transmission. Each device periodically generates a new key for distribution over the network.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NL	Netherlands
BE	Belgium	GN	Guinea	NO	Norway
BF	Burkina Faso	GR	Greece	NZ	New Zealand
BG	Bulgaria	HU	Hungary	PL	Poland
BJ	Benin	IE	Ireland	PT	Portugal
BR	Brazil	IT	Italy	RO	Romania
CA	Canada	JP	Japan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo			SE	Sweden
CH	Switzerland	KR	Republic of Korea	SK	Slovak Republic
CI	Côte d'Ivoire	KZ	Kazakhstan	SN	Senegal
CM	Cameroon	LI	Liechtenstein	SU	Soviet Union
CS	Czechoslovakia	LK	Sri Lanka	TD	Chad
CZ	Czech Republic	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	UA	Ukraine
DK	Denmark	MG	Madagascar	US	United States of America
ES	Spain	ML	Mali	VN	Viet Nam
FI	Finland	MN	Mongolia		

CRYPTOGRAPHIC APPARATUS AND METHOD FOR A DATA COMMUNICATION NETWORK

The present invention relates to a method and apparatus for encrypting and transmitting data.

It is well-known to transfer data between computers or other hosts over data communication channels. Such hosts are arranged in networks and allow transfer of data between a specific pair of hosts or to all hosts in the network according to established protocols.

It is often desirable to transmit sensitive data on such a network and therefore the network must be made secure, for example by limiting access to the hosts. Alternatively, data may be encrypted prior to transfer so that even if access to the network is obtained, any data intercepted is not meaningful.

Various techniques are known for encrypting data, many of which require a correlation between an encrypting operation performed on the data as it is generated and a decrypting operation performed by the recipient. To achieve this correlation, it is usual to provide keys that are used in a mathematical operation performed on the data. Some techniques, such as public key systems, utilize different keys at the sender and recipient but then require multiple transfers to achieve a secure transmission. Such a technique, however, reduces the overall data transfer capacity of a network and would not be compatible with existing communication protocols.

Other techniques utilize an identical but secret key at each host in the security domain. This permits data to be transferred with a single transmission. However, to provide the necessary degree of security, it is preferable to change the key periodically so that a prolonged observation of the encrypted data sufficient to yield the key is not possible.

The provision of a new key to each of the hosts in a domain must be accomplished in a secure manner; otherwise, the encrypted data becomes vulnerable. It has been proposed to distribute keys manually, i.e. utilize secure couriers to provide a new key to users of the network but this is time-consuming, expensive and provides too long a period between updates.

An alternative technique is to generate a key from a central unit and transmit it over the network. This, however, requires the transmission to be secure and requires the central unit to be operational at all times. A failure of the central unit disables the generation of new keys and may render the domain vulnerable.

It is therefore an object of the present invention to provide a method and apparatus for encrypting data that is compatible with existing communication protocols and may be utilized within a network environment.

It is a further object to provide a method and apparatus for generating keys on a periodic basis for use in the network.

In general terms, the present invention provides a security device at each host in a security domain which uses a key in combination with a specific mathematical function to provide an encrypting bit stream as data is received. The bit stream is then used in an encrypting function to encrypt and decrypt data as it is received. As the encryption proceeds, the key is modified by the mathematical function to generate the encrypting bit stream.

In the preferred embodiment, the mathematical function includes a register containing a secure, secret bit sequence. The key is used to generate an address for the register and extract the contents of the register for use in the mathematical function.

To transmit new keys, each device may generate a data packet that is used in part as the key and in part as the data in an encryption process. The resulting encrypted data is then used as a new key. Because the new key has been generated using a "secret" peculiar to the security domain, the key will also be peculiar to that domain. In this way, keys may be transmitted without encryption but still provide distinct unpredictable keys in a particular domain.

An embodiment of the invention will now be described by way of example only with reference to the accompany drawings, in which

Figure 1 is a schematic representation of a network having a plurality of security domains;

Figure 2 is a representation of a security device used in the network of Figure 1;

Figure 3 is a schematic representation of the operation of the device shown in Figure 2 to encrypt data;

Figure 4 is a schematic representation of the format of a packet distributed on the network of Figure 1;

Figure 5 is a schematic representation of the operation of the device shown in Figure 2 using the packet of Figure 4;

Figure 6 is a schematic representation, similar to Figure 3, of an alternative embodiment; and

Figure 7 is a representation of the operation of the embodiment of Figure 6 similar to Figure 5.

GENERAL NETWORK ARRANGEMENT

Referring to Figure 1, a local area network 10 in Figure 1 comprises a data channel 12 to permit the transfer of packets of data between a plurality of host computers 14, such as a computer or computer terminal. Individual hosts will be identified with alphabetic suffixes, i.e 14a, 14b, etc. Each of the hosts 14

requiring a cryptographic facility is connected to the data channel 12 through a security device 16 which is operable to encrypt data transmitted by a host 14 or decrypt data received by the host 14. Those hosts not requiring encryption, such as, for example, 14g are connected directly to the channel 12.

Data is transmitted in the channel in frames consisting of a preamble of a particular sequence of bits followed by a data packet. The packet consists of destination address (typically 48 bits), a source address, the packet length and the information to be transmitted. The information will be followed by a cyclic redundancy check (CRC) of the data transmitted on the channel 12. The format of the packet is described more fully below with reference to KEY DISTRIBUTION.

The first bit of the destination address will indicate whether the packet is to be broadcast on the network or is to be locally directed to a particular host. The exchange of data between the host 14 and the security device 16 and between the device 16 and the channel 12 is regulated by a conventional communications interface 15 operating on an established protocol as is well known in the art and will not be described further.

Each of the devices 16 performs a similar cryptographic operation on the data. However, to divide the network 10 into a plurality of distinct security domains 18a,18b, indicated by chain dot lines, the encryption keys used in the devices 16 of each domain 18

are different. Thus, encrypted data may be sent between hosts 14 of the same domain and will be received by hosts of different domains. However, these other hosts will not decrypt the data correctly.

Each of the devices 16 operates in a similar manner and therefore its operation in encrypting and decrypting data will be described in detail first. Thereafter, the interaction of the devices within the network will be described.

THE ENCRYPTION PROCESS

Referring to Figure 2, each of the devices 16 includes an encryption module 20 having a key register 22 which stores a 128 bit encryption key. At any given time, the key is identical in each operable device 16 in the same security domain 18. As will be explained, the key will be changed periodically within the domain and will also change as the encryption proceeds. Because the key is changed periodically, a 32 bit key sequence number is associated with each key and stored in a register 25.

The encryption module 20 operates under the control of the interfaces 15 to intercept data flowing between the host 14 and data channel 12 to perform an encryption process so as to encrypt and decrypt the data as it passes through the device 16. The device 16 also includes a key generator module 33 that is used to change periodically the key in the register 22 in a manner to be described below.

As best seen in Figure 3, the key stored in register 22 is used by each of a pair of parallel processing paths in each device 16 indicated as "B" and "L". However, only one path will be described in detail, as the processing is identical in both.

The bits of register 22 are initially transferred into an active register 23 where they are subdivided into discrete groups to provide 16 8-bit addresses, A_0 to A_{15}. Each address A contains an 8-bit word formed from 8 successive bits of the key in register 22.

A 1 x 256 bit register 24 is associated with each of the addresses A which together provide a primary memory 25. Each register 24 uses a respective one of the 8-bit words from the active register 23 as its read address. The 256 bit sequence in each register 24 in the primary memory 25 is maintained secret prior to and after installation. In general, the sequence in a register 24 is different to any other bit sequence in the other registers 24 of the same primary memory 25. In particular, the bit sequence in the register 24 in path B associated with address A_n will, in general, be different to the corresponding register 24 in path L. However, the bit sequence in corresponding registers 24 in different devices 16 in the same domain 18 will be the same. Thus, the bit sequence in register 24 in path B associated with address A_0 in device 16a will be identical to the bit

sequence in register 24 in path B associated with address A_0 in devices 16c,16d.

Each register 24 outputs a single bit contained at the address corresponding to the bit sequence in the associated address A so that a total of 16 bits are outputted by registers 24 of primary memory 25. The 16 bits are grouped into two 8-bit words and each is used as the address for a respective 1 x 256 bit register 26 which together form a secondary memory 29. The bit sequence in each of the registers 26 in the same device is in general different and secret. However, it is identical to the corresponding register 26 in the secondary memory 29 in all other devices 16 of the same domain 18.

Each of the registers 26 outputs a single bit indicated as P,Q corresponding to the address designated by the 8-bit word. A pair of PQ bits is similarly generated from the parallel processing path L and each pair of bits is applied to a switch 27 which selects one of the pair of P,Q bits. If the destination address indicates that the packet is to be broadcast, the output of path B is selected, and conversely a local destination address ensures that path L is selected. The bits selected by switch 27 are applied to an exclusive OR function 28 which generates a single output bit identified as FEK and used to encrypt incoming data.

The FEK bit is applied as one input to an exclusive OR (XOR) function 30. A bit of the data stream

received at the device 16 from associated host 14 to be encrypted is applied to the other input of XOR function 30 so that the output is encrypted data that is the product of the exclusive OR function of the FEK bit and the data. The encrypted bit is then transmitted from the device 16 to the channel 12.

The selected P and Q bits are also applied to a 2 x 4 truth table 32. Table 32 produces a different 4-bit output for each combination of P and Q. Thus, if P and Q are both 0, the 4-bit output may be 1010, whereas if P is 1 and Q is 0, the output may be 0110. It is preferred that output combinations having two 1's and two 0's are used.

The output of table 32 is replicated 4 times and distributed through a 128-bit sequence that is stored in an adder 34. Adder 34 is used to increment the active register 23 so that each 8-bit cell of the register 23 will be incremented by one bit of the output of table 32. Thus, if the output of truth table 32 is 0110, the bits in address $A_{0,3,4,7,8,11,15}$ will not be changed but the bits in address $A_{1,2,5,6,9,10,13,14}$ will be incremented by 1. In the event that an address A_i that has all 1's is incremented, its value will reset to all 0's with no overflow. Provided output combinations with an equal distribution of 1's and 0's are used in Table 32, the bit sequence in half of the addresses A in the key will change and produce a new key in the active register 23.

The generation of a new FEK bit then proceeds using the new key in active register 23, and is XOR'd with the next bit of data. The entire packet is encrypted with successive FEK bits in this way, starting at the second bit of the destination address and proceeding until the last bit of the CRC, which is the last bit of the packet. The first bit of the destination address is not encrypted. It is used to indicate whether the B path ("1") or the L path ("0") was used to generate the FEK bits.

Finally, the encrypted packet has appended to it a CRC field so that the encrypted packet will seem normal to any computers, such as 14g of Figure 1, that does not connect to the network 12 through a security device 16. This extra CRC has no other purpose. It is ignored by the security devices 16 that receive the frame containing the packet.

The encrypted frame is composed of a preamble followed by the encrypted packet and the appended CRC. This encrypted frame is transmitted on the channel 12 in the normal way that unencrypted frames are transmitted.

The same process is used to decrypt the data as it is received by a security device 16 in the same domain on the channel 12. If a meaningful decryption can be made, the key in register 22 will correspond to the key first used to encrypt a bit of the packet. The FEK bit initially generated by the key will therefore correspond to the FEK bit initially used to encrypt the data. By

XOR'ing the incoming encrypted data bit with the same FEK bit, the original data is obtained. Since the XOR function is reversible by a second encryption with the same key, the original bit stream results after the decryption provided the contents of the secret registers 24,26 are the same in the decrypting device as they were in the encrypting device.

An attempt to decrypt the data packet with a device 16 from another domain will not succeed as the bit sequences in the registers 24 and 26 will differ. Thus, the same key will produce a different sequence of FEKs and will not correctly decrypt the data. This will be apparent from a comparison of the CRC included in the encrypted data and that obtained after decryption.

After the encryption and decryption has been successfully completed for a packet, each 4-word span of the key stored in register 22 in each of the devices 16 is incremented by the key sequence number stored in register 25. The newly generated key is then transferred to the active register 23 upon receipt of the next packet.

Thus many packets may be encrypted from the same initial key with the key changing bit by bit within a packet and also changing on a packet-to-packet basis.

OPERATION WITHIN A NETWORK

The destination address of a data packet will indicate whether the data is to be broadcast to each host

14 within a domain or to a specific host, i.e. local transfer, within the domain.

Where the data is to be broadcast, each host 14 within the domain will receive the data decrypted by its associated device 16 and, upon completion of the packet, will update the key in register 22. Devices 16 associated with hosts outside the domain will also receive the data packet but an attempt to decrypt the data will result in an incorrect CRC because the secret in their broadcast path B is different.

There is an important difference in the operation of the security device 16 when it is decrypting a packet using the L path rather than the B path. In this case, the decrypted destination address that is generated bit-by-bit is compared bitwise with the network address of the associated computer 14 to which the frame containing the decrypted packet is being sent. As this bitwise comparison is taking place, the first 47 bits of the destination address of the associated computer replace the actual destination bits of the packet and are forwarded to the computer in their place.

If the result of the bitwise comparison shows that the decrypted destination address is exactly that of the associated computer 14, the 48th and last bit of its own destination address is sent to the computer, and the decrypted packet is sent bit-by-bit to the computer up to but not including the CRC that was added during encryption.

If the result of the bitwise comparison shows that any bit of the decrypted destination address in the incoming packet differed from the corresponding bit in the network address of the computer, the 48th and last bit of the destination address that is sent to the computer is the complement of the 48th bit of the last bit of its network address, and a standard pattern of bits, one pattern bit for each incoming packet bit, is forwarded to the computer 14 instead of the decrypted packet. The standard pattern is followed by an appropriate and correct CRC at the point where the CRC appears in the packet before encryption. The standard pattern is preferably chosen to make this CRC easier to compute. Once the transmission is complete, the key register 22 in all devices that have seen a complete packet with a correct CRC 16 are updated as previously described. In this way, the key in the register 22 remains the same for each device 16 in a domain even though the data packet is only decrypted by one device in the domain.

If for some reason the transmission is interrupted, no change is made to the key in register 22.

As noted above, each encrypted packet includes a CRC of the data as transmitted. This permits hosts without a device 16, such as host 14g, to process the packet through its interface and also permits the distribution of unencrypted packets throughout the network as may be desirable but only among those

computers not attached to the network through a security device 16.

KEY DISTRIBUTION

To utilize the above encryption process, it is necessary to generate an identical key in each device 16 in the same domain 18. It is also preferable that the key in each domain is different. The encryption process described above is utilized in the periodic generation of a new key within a domain as shown in Figures 2 and 4.

The key generation module 33 in each device 16 is used to generate periodically a key distribution packet that is transmitted over the data channel 12 and processed by the devices 16 in the same domain to generate a new key in a manner to be described below assuming that one of the devices 16 has control of the channel 12 in a collision-free manner.

The key distribution packet must be compatible with normal data packets and therefore has a similar format. However, indicators within the packet are used to identify it to other devices 16 as a key distribution packet and ensure that it is processed by the module 33 to generate the new key.

The format of a data packet produced by the key generator module 33 is shown in Figure 4. Each packet has a minimum bit length (in this example, 512 bits) and is arranged in notional blocks of bits. The packet will of course be preceded by a preamble as is usual.

The first two blocks of the packet, whether used for normal data transfer or key distribution, are each 48 bits and are respectively the destination address block 35 and the source address block 36. When normal data is to be transmitted, each address block 35,36 will be a 48 bit code indicative of the destination and source computer respectively. The first bit of destination address block 35 indicates whether or not the packet is to be processed by the broadcast path B, indicated with a "1", or by the local path L, indicated with a "0". The second bit of the addresses 35,36 is used to indicate whether or not the frame identification is under local control (1) or is a worldwide unique code (0). With a normal data packet, worldwide unique codes will be indicated and followed by a 46-bit host computer address.

Where the data packet is directed to a specific address, the destination address block 35 would commence with 00 or 01 and is followed by a 46 bit address for the recipient host computer 14. However, if the data packet is to be broadcast over the network, the destination address block 35 would be constituted by a 1 followed by 47 bits, typically all "1's".

A source address block 36 follows the destination address block 35 and is used to indicate the origin of the data packet. The source address will always begin with an "00" or "01" and will be followed by a 46 bit identifier code uniquely identifying the source computer.

To distribute a new key in a security domain, a key distribution packet (KDP) is generated by module 33. Module 33 has special address contents for both destination address and source address stored in an address register 51 in the module 33 and is recognized as a key distribution packet by these contents. The destination address of a KDP could be 48 "1"s, indicating a broadcast packet but is preferred to use a special destination address peculiar to a KDP. The source address 36 of a KDP is a specific bit pattern, beginning with "00" and followed by a 46-bit identifier code reserved for this purpose. The 48 bits of the destination address and 46 bits for the source address can be chosen during device manufacture. They will, however, be worldwide unique to a particular network; that is, all the devices 16 attached to the network and intended to change their keys synchronously will be identified with the same code. Typically, this will be limited to single domain.

The worldwide uniqueness of the identifier codes is assured because these bit combinations are regulated by an international organization such as the IEEE. In the present case, the identifier code used as the destination and source address will be indicative of a message originating at a security device 16 and therefore is a prime indication that the packet is a key distribution packet. Therefore, by using the 48-bit destination and source addresses, it is possible to

distinguish a key distribution packet from a normal data packet.

The packet length is indicated by a 16-bit packet length block 37 which, with normal data packets, precedes the information to be transmitted. When the packet is a key distribution packet, however, the packet length block 37 is followed by a 32-bit key sequence number block 38 derived from the key sequence register 25 in encryption module 20 of the device 16 generating the KDP. The data in key sequence number register 25 is incremented by 1 for insertion in block 38 so that as the keys are updated in a domain 18, the key sequence number used in that domain changes in a controlled manner. Each host computer in the same domain should be operating with the same key sequence number.

The key sequence number block 38 is followed by a 96 bit data block 40 and a 128-bit data block 42 separated by a fixed length padding block 45. The data in blocks 40,42,45 is generated pseudo-randomly by a random number generator 53 in the key generator module 33 in the security device 16 that originated the KDP.

The next data block 43 is 128 bits long, also obtained from the random number generator 53 in the security device 16 that originated the KDP. It is followed by a fixed length padding block 45 and a data field called the CRC frame integrity block 44 derived by the transmitting security device 16 during transmission of the KDP.

As the key distribution packet is transmitted, the CRC frame integrity block 44 has to be generated and this is done by utilizing the encryption module 20 in the generating security device. As shown in Figure 4, the key sequence number 38 and the data block 40, totalling 128 bits, are loaded into the key register 23. The normal FEK encryption mechanism 20 is used, with the B path selected by switch 27, to encrypt the incoming 128 bit data block 42 and to place the 128 bit result into a holding register 50 in module 33. The contents of register 50 are then transferred to register 23, and this new key is used to encrypt data block 43. A cyclic redundancy check (CRC) of these encrypted bits is performed as the encryption proceeds and is used as the transmitted frame integrity block 44.

Data blocks 40, 42, 43 and 44 are separated by the fixed length fields 45 to allow time for processing in the originating and receiving security devices 16.

The packet is completed by a padding block 46 to satisfy the minimum length requirements of the protocol and a 32 bit CRC block 48 that is generated from the bits of the packet as transmitted to check for error-free transmission.

KEY GENERATION

The key generator module 33 is activated to transmit a key distribution packet under the control of a timer 62 and attempts to gain access to the line 12.

Assuming that access is obtained, the key distribution packet is generated and transmitted over link 12 to be received at other security devices 16 on the channel 12. It is identified by each of the security devices 16 as a KDP because of the bit combination of the broadcast destination address 35 and the source address block 36. The key sequence number in block 38 is compared with that in register 25 in the receiving security device 16 and if the new key sequence number is greater than the existing one, the production of a new key proceeds. If the key sequence number is not greater than the existing one, the KDP will be ignored. Throughout the generation of the key, the device 16 will maintain a data stream to the associated host 14 to prevent generation of new packets from the computer interfering with the key generation.

A similar process is followed to that used to generate the CRC frame integrity block 44 to generate a new key. At each of the receiving devices 16 having the correct correlation of key sequence numbers, the new key sequence number block 38 and data block 40 are loaded into the register 23 and used as the active key to encrypt the data block 42. The resulting encrypted data is stored in the register 50 as the potential new key which should correspond with the contents of the register 50 in the generating device 16. To check for the integrity of the transmission and encryption, the contents of register 50 are transferred to the active register 23 and used as the key to encrypt the data block

43. As the encryption proceeds, a CRC is performed on the 128 bits of the block 43 and the result compared with the frame integrity block 44. If these are identical, the transmitting and receiving security devices must have the same B path memories 24,26 contents, and assuming physical security is adequate, the receiving device 16 is in the same domain as the generating device 16 and the contents of register 50 are transferred to the register 22. The key sequence number in register 25 is also replaced by the new key sequence number and each device in the same domain is operating with a new but identical key. Thereafter, normal data may be transferred within the domain.

It will be noted that although the key generator data packet is broadcast throughout the network 10, a secure key is generated in each security domain by virtue of the unique secure bit sequences used to generate the FEK bit. If a packet appears to be a KDP but its key sequence number is not greater than the sequence number of the key it is using, or if a packet appears to be a KDP but the integrity block 44 does not satisfy the comparison described, or if the packet appears to be a KDP but the overall packet CRC does not work out as it should, then some flaw has been detected in the packet. In all cases, invalid key sequence number, invalid integrity block 44, or invalid packet CRC, the packet is ignored and no change is made to the key register 22 or key sequence number 25.

PERIODIC GENERATION OF KEY DISTRIBUTION PACKET AND INITIALIZATION

The above process assumes that a key distribution packet is received periodically and it is a particular benefit of the present system that each of the security devices 16 has the capability of generating a key distribution packet. It is, however, possible to provide a central control to generate such packets if preferred.

Because each of the devices 16 has the capability of generating a new key, the timer 62 in key generation module has a variable countdown period to ensure that one device 16 does not monopolize generation of the key distribution packets. After a particular device has gained access to the channel 12 and transmitted a new key, timer 62 of that device 16 is reset to an initial period $(60-j\delta)$ seconds where δ is an arbitrary time, e.g. 100 μs, and j is an integer that initially is zero.

Each time a new key distribution packet is received by the device 16 and processed by the key generation module, a signal is also applied to the timer 62 to increase the value of j by 1, i.e. decrease the interval set by the timer 62. Thus, as other devices 16 generate key distribution packets, the interval set by timer 62 will progressively decrease and ensure that eventually its associated device 16 will gain access. Of course, once access is obtained, a reset signal resets timer 62 to the initial maximum countdown period. If a

collision is detected during transmission of the key generation packet, the timer 62 is reset to its previous value. This preserves the status of the device 16 in the key generation process and also avoids progressively decreasing intervals between unsuccessful attempts. This would tend to cram a network being utilized near its maximum capacity.

Because hosts 14 can be connected or disconnected from the network at any time, it has to be recognized that after connection or reconnection, an initialization period is required before the device 16 will operate with the same key as other devices in the same domain. To mitigate the possibility of a newly-connected host 14 from gaining access and generating a key, the timer 62 is conditioned to calculate an increasing time-out interval. At initialization, timer 62 sets the timeout counter at $(60+j\zeta)$ seconds where ζ is a function of the serial number of the unit 16 in which the module 33 is located. This will always be greater than the interval set by the counter 62 of a previously connected host 14 and therefore a new key will be received before the timer of the new device counts down. When receiving the new key distribution packet, the discrepancy between key serial numbers is ignored and, provided the CRC frame integrity blocks 44 are the same, indicating a common domain, the transmitted key serial number is adopted and entered in the register 25.

GENERAL OPERATION

In operation, therefore, a host 14 that wishes to transmit data over the channel 12 will monitor network usage through the interfaces 15.

According to its normal protocol, it will choose a time to transmit the frame containing the packet so as to minimize interference with other such frames. As the frame is received in the security device 16, it is encrypted by the encryption unit 20 and transmitted on the network 12. The first bit of the destination address is not encrypted so that when it is received, it can be identified as a broadcast packet, encrypted through the B path of the unit 20 or a local packet, encrypted through the L path of the FEK unit.

The received data is decrypted by the decryption unit 20 and its destination address 35 examined to determine if it is intended for the associated host 14. If it is, the decrypted data passes through the communications interface 15 to the host 14. If the destination address indicates that the message is not intended for the associated host 14, the packet is ignored but the interface unit 15 maintains a data stream to the host to prevent the host requesting access to the link 12 and initiating a collision.

Data received by devices 16 or unprotected hosts 14 outside the security domain of the originating host 14 will not be able to decrypt the data as the keys and the contents of registers 24,26 will differ.

When one of the timers 62 finishes its countdown, a signal requesting access to the channel 12 is sent to the interface 15. If the data channel 12 is busy, the access is refused and the timer is reset to its previous value. If the data channel 12 is available, the generation of a key generation packet is initiated and transmitted over the channel 12. Devices 16 in the same domain will recognize it as a key generation packet and proceed to generate a new key as detailed above. Once transmission has been completed, the timer 62 is reset to the maximum period and the timers 62 in each of the other devices are rest to a reduced interval.

It will be seen, therefore, that the network 10 is self-sustaining in that new keys may be periodically generated by any of the security devices 16. The generation of different keys in each security domain enables a key generator packet to be broadcast throughout the network from any of the devices without compromising the security. The integrity of the domain is maintained by ensuring that the devices 16 are tamperproof and do not require modification of the hosts 14 or data channel 12. The encryption algorithm ensures progressively varying encryption keys that are periodically changed and therefore, in practical terms, entirely secure.

ALTERNATIVE CONFIGURATIONS

As described above, the secrets in registers 24 and 26 in broadcast path and local path of domain are

different to those of other domains. This means that the keys are generated on a domain-by-domain basis. In certain instances, it may be desirable to have a common key throughout a network but still retain separate domains. The present arrangement has the flexibility to accomodate this by providing common registers in the broadcast path of all domains but retaining differences between domains in the local path. The key distribution packet will then be recognized and processable by all devices 16 in the network to generate a common key provided its CRC block 44 is derived from an encryption in the B path. Even though a common key is used, secure transmission can still occur within the domain by using the local path.

SECOND EMBODIMENT

A further embodiment of the security device 16 is shown in Figures 6 and 7, with Figure 6 schematically illustrating its operation during encryption and decryption, and Figure 7 illustrating the generation of and distribution of new keys. Components having a similar function to those described in the embodiments of Figures 1 through 5 will be identified with like reference numerals with a suffix "a" added for clarity.

FEK BIT GENERATION

Referring therefore to Figure 6, an active register 23a is formed from three linear recurring

sequence registers (LRS) each of which contains a portion of the key. The LRS register is a commercially available register having the facility for internal feedback connections between cells of the register and may be arranged to ensure that no repetition of the sequence within the register occurs within 2^{32} bits. Such registers are readily available.

A first register 70 is identified as the key distribution frame (KDF) register and contains the key distribution frame sequence number that is transmitted with a key distribution frame as will be described in further detail below. The second register 72 is identified as a successful frame count register (SFC) and its contents are initially derived from the transmission of a key distribution frame. The contents of the SFC register 72 are incremented after a frame has been transmitted and for the purposes of the protocol, it is assumed that the transmission of 512 bits indicates that a frame has been transmitted successfully. The count in the SFT register 72 is incremented after each frame.

The third register 74 is identified as the FEK bit count register (FBC) and its contents are also generated during distribution of a key distribution frame. The contents of the FBC register 74 are incremented by 1 after each generation of a FEK bit so that during transmission of a frame, the contents of the FBC register 74 are continuously changing. A backup register 76 stores the initial value of the FBC register

74 and reloads it after the transmission of each frame. This is necessary as the contents of the FBC register will be incremented even if the transmission of the frame is terminated due to a collision with another frame and so the contents of the register 74 in one security device would differ from that in other security devices in the same domain. Accordingly, by reloading the initial value of the FBC register at the start of each frame, all devices in the domain will have the same contents for the active register 23.

The contents of the active register 23 are used to derive an address for each of the columns in a primary memory 25a. Each column 24a of memory 25a corresponds to a register 24 shown in Figure 3 and provides a single bit output for each address. The address for each column 24a is derived from a 96 bit address register 78 which receives the bits of each of the registers 70,72,74. The bits of the registers 70 through 74 are interleaved in the primary memory address 78 such that two bits from the register 70 are followed by two bits from the register 72 which in turn is followed by two bits from the register 74. Six such bits are then grouped to provide a six-bit address for a respective register 24a. Each of the columns 24a has a distribution of 1's and 0's which is approximately equal and each of the columns 24a has a combination that is secret and preferably different from any other column in the primary memory 25a. As before, however, each of the columns 24a in one of the devices 16

will have a corresponding register 24a in another of the devices 16 in the same domain and sharing the same secret.

The output from the primary memory 25a is a 16-bit address which is used to address a 1×2^{16} secondary memory 29a. The contents of the secondary memory 29a are also secret but identical with other devices in the same domain and have a substantially equal distribution of 1's and 0's within the memory. The output from the secondary memory 29a is a single FEK bit which is exclusive OR'D at the XOR function 30a with incoming data. Encrypted data is then transmitted as the outgoing bit stream.

Upon successful transmission of 512 bits, a frame detector 80 assumes that a successful frame transmission has occurred and, upon detection of the start of the next frame, will increment the contents of the SFC register 72 by 1. The detection of a start frame delimiting pattern in the preamble of a frame will also reload the contents of the FBC register 74 so that the initial address in the primary memory address 78 will differ from frame to frame. The contents of the KDF sequence number register 70 remain constant until such time as a new key is distributed over the network. It will be seen, however, that the FEK bit is generated from a dynamic key to generate the addresses for two memories, the contents of which are secret.

KEY GENERATION

The generation of the key is again accomplished by any of the security devices 16 as will be described with reference to Figure 7. The format of the frame distribution packet is generally similar to that shown in Figure 4 and includes a preamble followed by a start frame delimiter sequence followed by a destination address 35a. The destination address 35a indicates that the frame is to be broadcast within the domain and is followed by a source address 36a. The source address 36a is derived from a register equivalent to the address register 51 in Figure 2 and identifies to the recipients of the frame that the frame is a key distribution packet. The recognition of the source address 36a causes the contents of the registers 70,72,74 to be temporarily stored in parallel registers 70a,72a,74a so that if the frame is not correctly received, the previous contents of the active register 23a can be restored.

The recognition of the source address 36a also initializes the LRS 70,72,74 so that they contain a full count of 1's. A pad 37a is provided between the source address and the KDF sequence number 38a to allow for the initialization of the registers. The KDF sequence number is the contents of the KDF register 70 incremented by 1 and is initially compared with the existing contents of the register 70 to ensure that it is a valid key sequence number. Assuming that it is, the new key distribution sequence number 38a is loaded into the KDF register 70.

The KDF sequence number 38a is a 64-bit sequence which is loaded into the register to initialize a new sequence of bits in the register. This sequence will be identical for each device 16 in the same domain. The contents of the registers 72 and 74 are still all 1's.

A pad is provided after the KDF and is then followed by a data field 40a. The data field 40a is a random sequence of 64 bits generated by the random number generator 53 in Figure 2. This sequence of bits is fed through the XOR 30a and encrypted by a sequence of FEK bits produced using the contents of the registers 70,72,74 to generate the addresses for memories 25a and 29a.

The first 32 bits of encrypted data are fed into the SFC register 72 to generate a new SFC. At the same time, the generation of a FEK bit from the secondary memory 82 also increments the FBC register 74 so that its contents are changing as the first 32 bits are fed through the XOR gate 30a.

The next 32 bits are also encrypted and are exclusive OR'D with the FEK bit count to increment the FBC register 74.

After the random data 40a has been processed, the frame distribution packet includes a pad 45a followed by a data field 42a made up of 512 bits generated by the random number generator 53. The purpose of the second data field 42a is to check the integrity of the frame distribution packet by means of the integrity CRC 44a

appended to the packet. This check is done, and indeed the ICRC 44a is generated using an ICRC generator 84 which is a 32-bit LRS register similar to that used for the registers 70,72,74. Each of the bits of the random field 42a is encrypted with the FEK bit by the XOR gate 30a and fed to the ICRC generator 84. This is initially set at a full count - that is, all 1's - and is incremented by the value of the encrypted bit. At the end of the data field 42a, the contents of the ICRC generator 84 should match those of the ICRC field 44a. The contents are compared, and if the patterns are identical, it is assumed that the frame distribution packet has been transmitted satisfactorily. The value of the FBC register 74 is then stored in the FBC register 76 and the contents of the registers 70,72,74 operate as the new key.

If for some reason the ICRCs do not match, the values in the registers 70,72,74 are deleted and the previous values temporarily stored in registers 70a,72a,74a are replaced until a new frame distribution packet is recognized.

It will of course be understood that the generation of a frame distribution packet is essentially the same with the destination data 35a, source data 36a, key sequence number 38a and the random data provided by the frame distribution module 33. Again, therefore, the encryption module 20a is utilized to generate a new key

both for transmission and for reception in each register in the same domain.

If the frame distribution packet is received in another domain, the primary memory and secondary memory 25a,29a will have different secrets and therefore will not generate a new key in which the ICRCs are matched. This is because the ICRC is generated using the encryption keys that are peculiar to a particular domain.

We claim:

1. A data communication system comprising a plurality of hosts interconnected by a communications channel to allow data transfer therebetween, at least some of the hosts having a cryptographic security device associated therewith to organize said hosts into a common security domain, said device having an encryption function operable to encrypt data transmitted through the channel to other hosts in said domain and decrypt data received through the channel from other hosts in the domain, a packet generating function to generate and distribute on said channel a key distribution packet, and a key generation function to receive said key distribution packet and generate therefrom an encryption key for said encryption function that is common to each host in said domain.

2. A data communication system according to claim 1 wherein said key distribution packet includes an identifier to distinguish said key distribution packet from other data packets and each of said devices includes means responsive to said identifier to direct said key generator data packet to said key generating function.

3. A data communication system according to claim 1 wherein said packet generating function in each device operates periodically to request access to said channel.

4. A data communication system according to claim 3 wherein said packet generating function includes a timer to determine the interval between requests for access to said channel.

5. A data communication system according to claim 4 wherein said timer is adjustable and intervals between requests progressively decrease until access to said channel is obtained.

6. A data communication system according to claim 5 wherein said interval is decreased upon receipt of a key distribution packet from another device and generation of a new key therefrom.

7. A data communication system according to claim 1 wherein said key distribution packet includes first and second portions, said first portion being utilized by said key generation function as a key in said encryption function to encrypt said second portion and thereby provide a new key for said encryption unit.

8. A data communication system according to claim 7 wherein said key distribution packet includes a check function obtained by encryption of a portion of said key distribution packet by said new key.

9. A data communication system according to claim 8 wherein said key generator packet includes a third portion to be encrypted by said new key to generate said check function.

10. A data communication system according to claim 9 wherein said first, second and third portions are generated by a pseudo random number generator within said packet generating function.

11. A data communication system according to claim 1 wherein each of said encryption functions has a key serial number associated therewith and indicative of the key utilized by said encryption function, said key distribution packet including an indicator derived from said key serial number and providing a means to associate devices utilizing a common key.

12. A data communication system according to claim 11 wherein said indicator is included in one of said portions.

13. A data communication system according to claim 6 wherein said timer is reset to a predetermined maximum upon a device associated therewith transmitting a key distribution packet to other hosts in said domain.

14. A data communication system according to claim 13 wherein said timer is set to a period exceeding said maximum upon connection of said device to said system.

15. A method of distributing an encryption key in a data communication network having a plurality of host interconnected by a data communication channel and each host having an encryption function associated therewith comprising the steps of transmitting on said network a key distribution packet, utilizing a first portion of said packet as a key in said encryption function, encrypting a second portion of said packet by said encryption function with said first portion utilized as said key and using the encrypted data as a new key.

16. A method according to claim 15 including the step of comparing a check function derived from said new key with a check function contained in said key distribution packet to confirm an identity of encryption function between a device generating said key distribution packet and a device receiving said key distribution packet.

17. A method according to claim 16 including the step of utilizing said first and second portions of said key distribution packet as a key and data respectively in said generating device to provide a new key, utilizing a third portion of said packet as data to be encrypted by

said new key, deriving from the encrypted data a check function and including said check function in said key distribution packet for comparison with a check function similarly derived at said receiving device.

18. A method of encrypting data for transmission on a communication channel comprising the steps of establishing a key having a plurality of bits, grouping selected ones of said bits to provide an address for accessing a register containing data, outputting the data contained at that address, utilizing the data in a predetermined manner to provide a bit to encrypt a corresponding bit of the data to be encrypted, and modifying the key in a manner derived from the generation of the encryption bit to provide a different bit sequence at said key to generate a different address for said register when generating the next encryption bit.

20. A method according to claim 18 wherein the output from said register is utilized as at least one bit of an address for a further register, the output from the further register being utilized to generate said encryption bit.

21. A method according to claim 18 wherein a plurality of discrete groups are formed from said key and a register is associated with each group.

22. A method according to claim 21 wherein outputs of said registers associated with said groups are organized to provide an address for each of a set of further registers, each which provides an output from which the encryption bit is utilized.

23. A method according to claim 22 wherein said set of further registers includes a pair of registers each of which provides a single bit output.

24. A method according to claim 23 wherein said bits are combined to produce said encryption bit.

25. A method according to claim 24 wherein said bits are combined by an exclusive OR function.

26. A method according to claim 22 wherein said outputs are utilized to determine the modification of said key.

27. A method according to claim 22 wherein the outputs determine which of said bits of said key are to be changed.

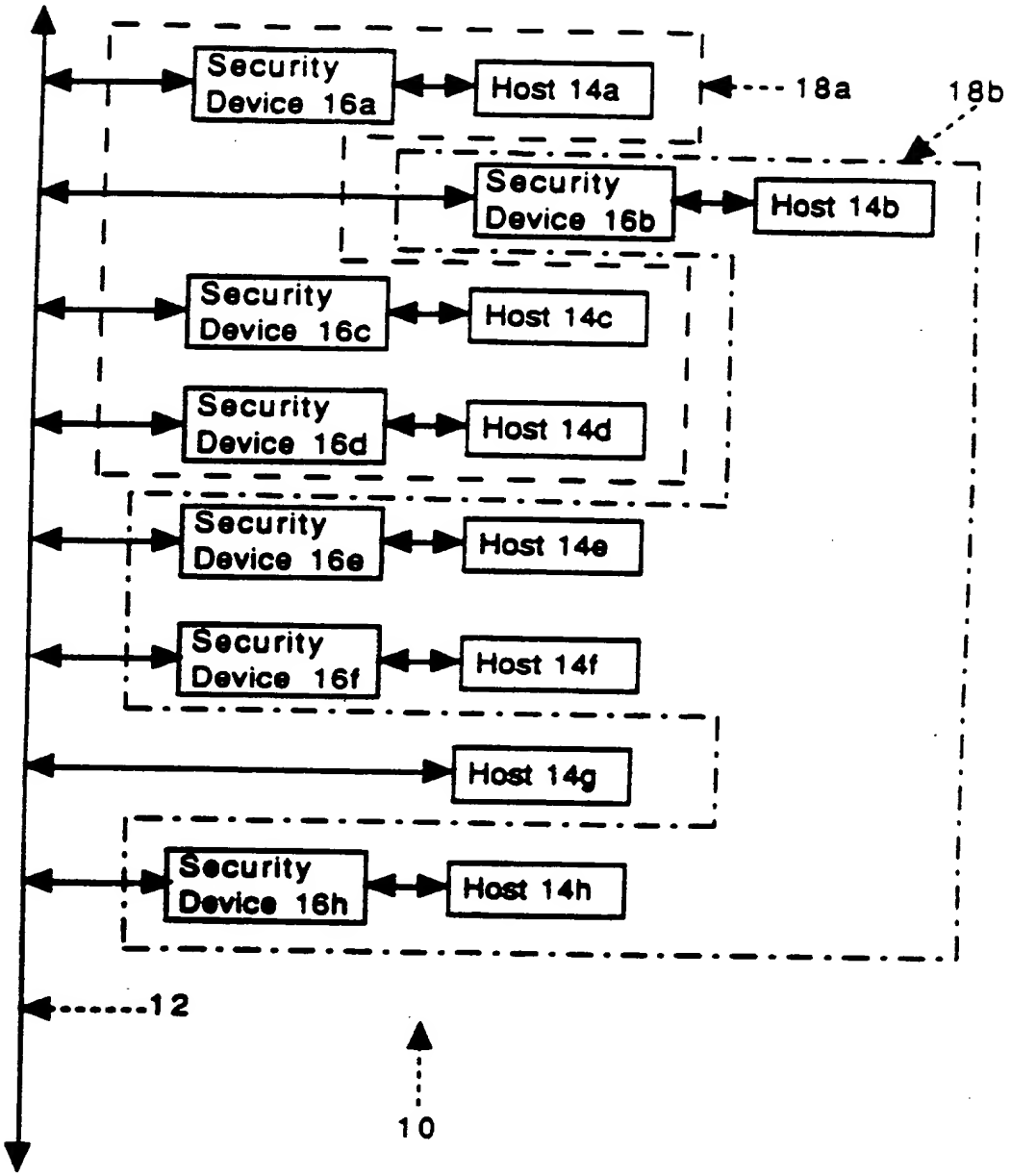


Figure 1.

SUBSTITUTE SHEET

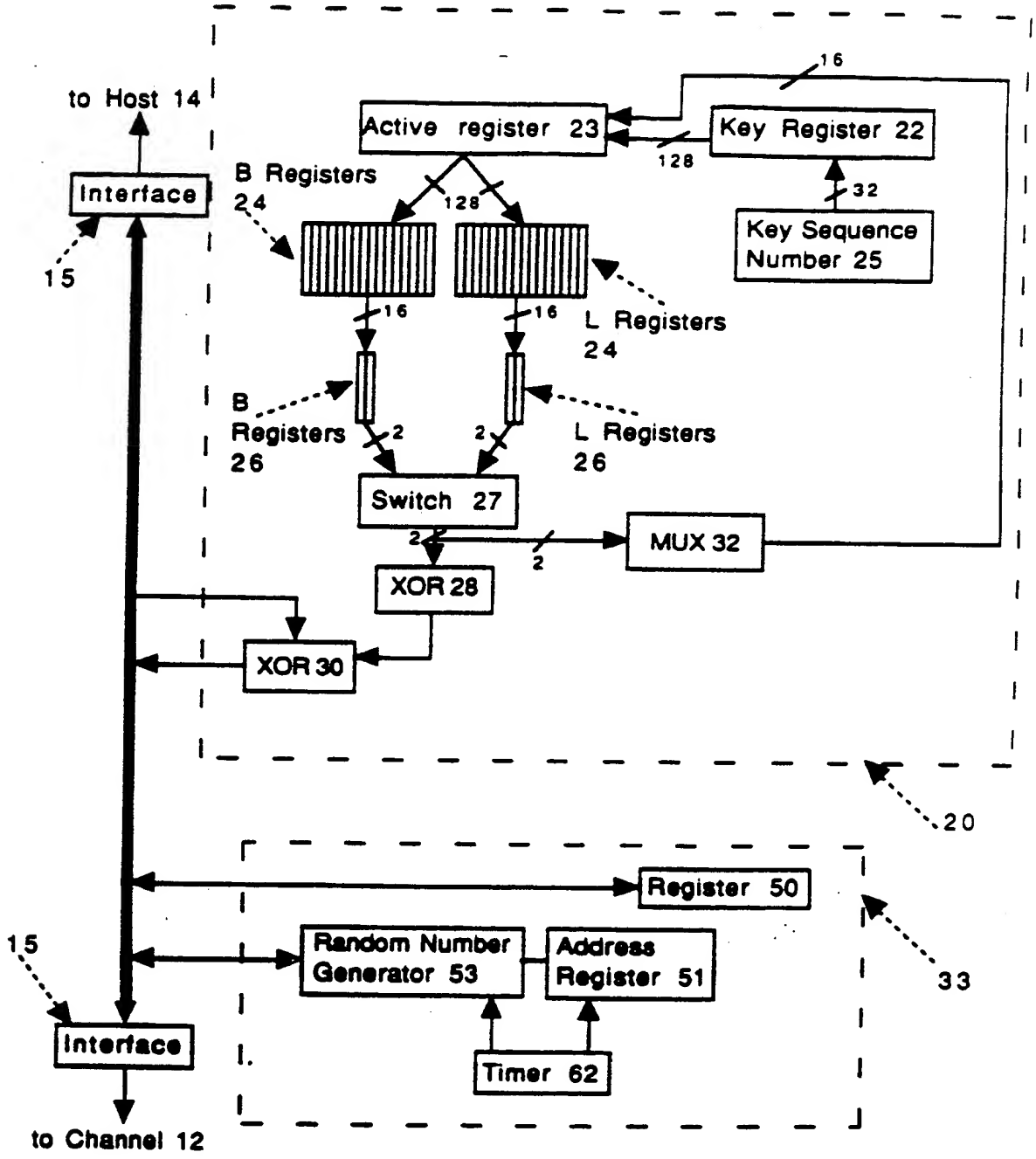


Figure 2

SUBSTITUTE SHEET

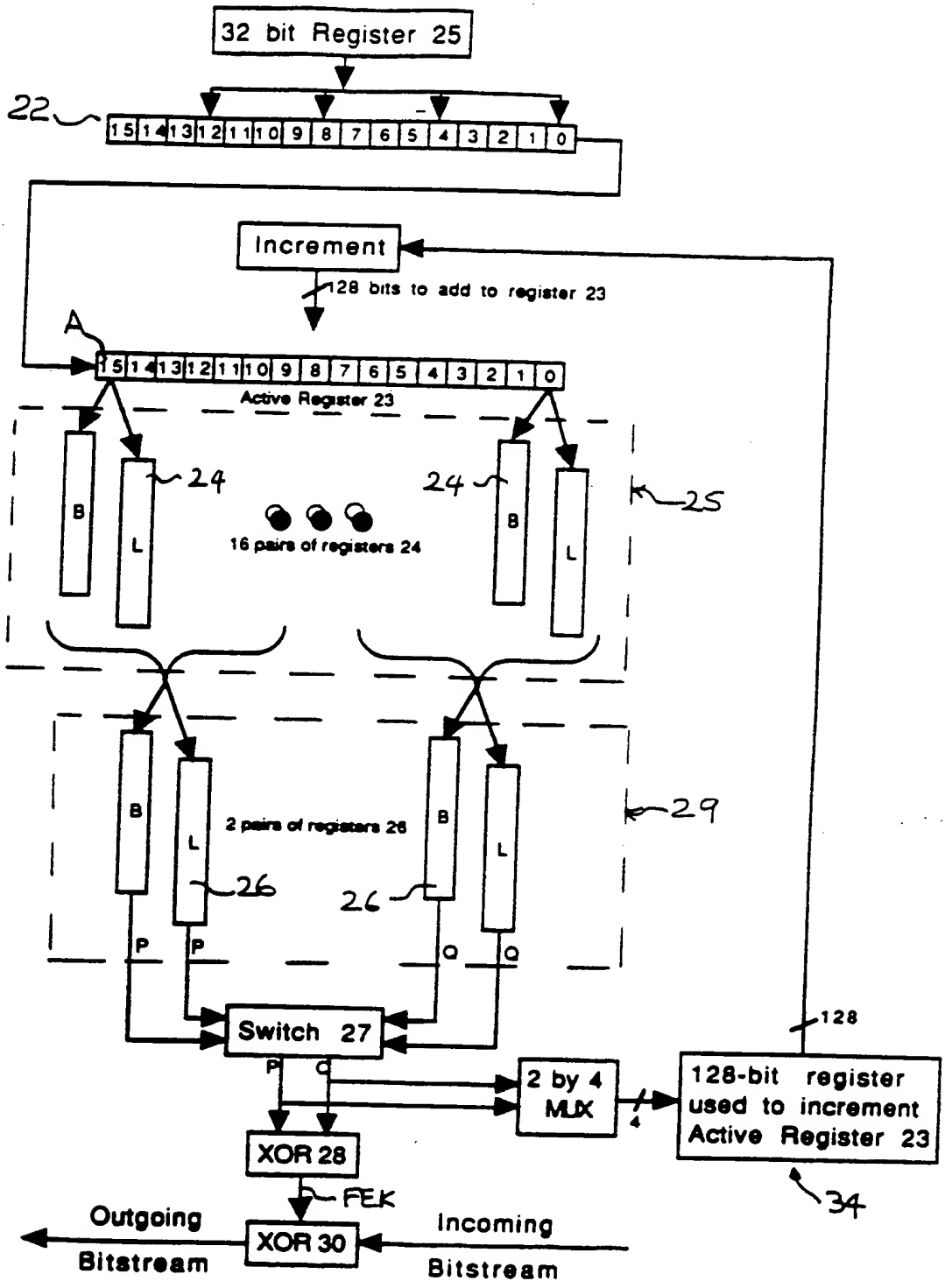


Figure 3.

SUBSTITUTE SHEET

FIG. 4

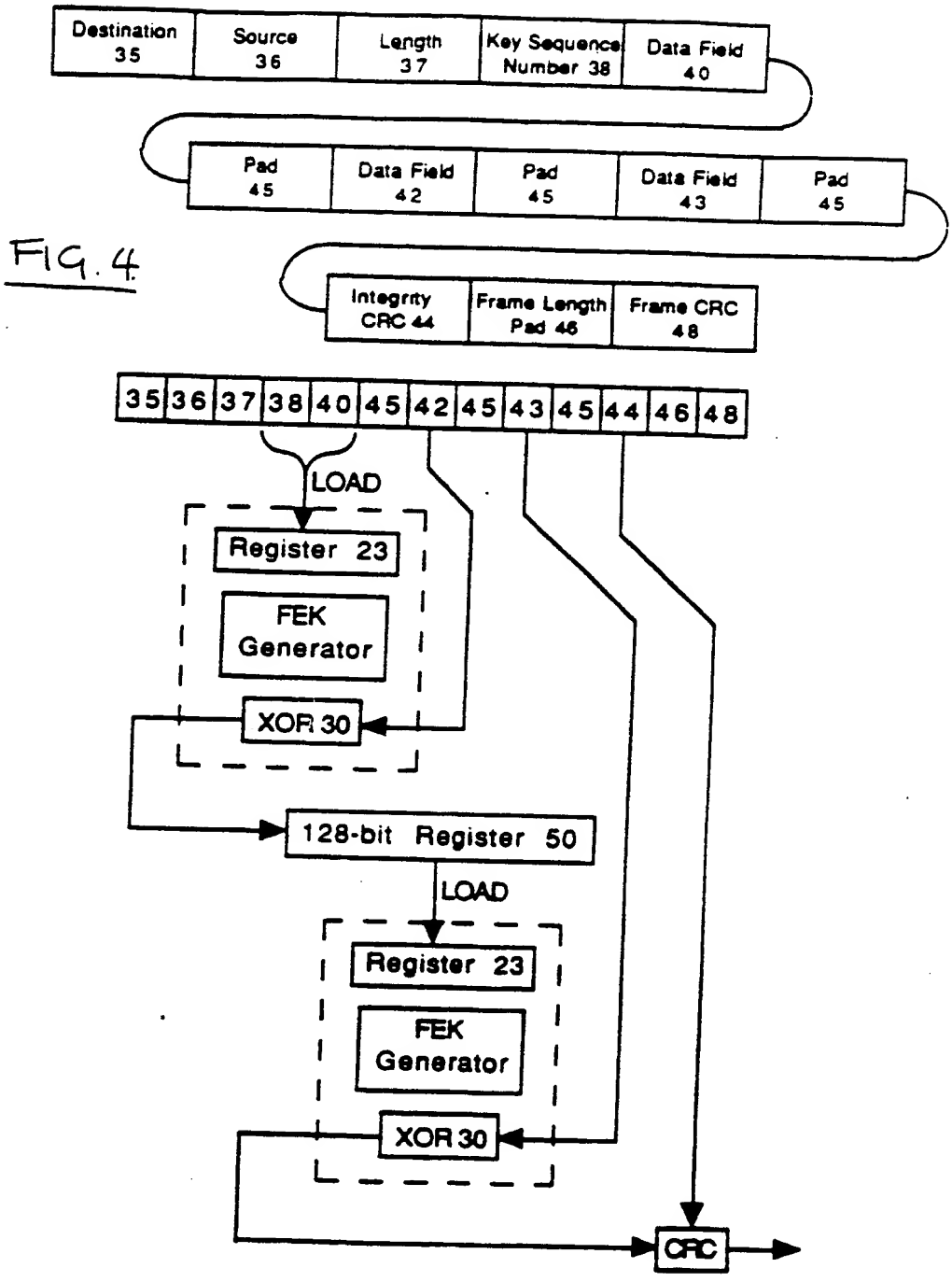


Figure 5

SUBSTITUTE SHEET

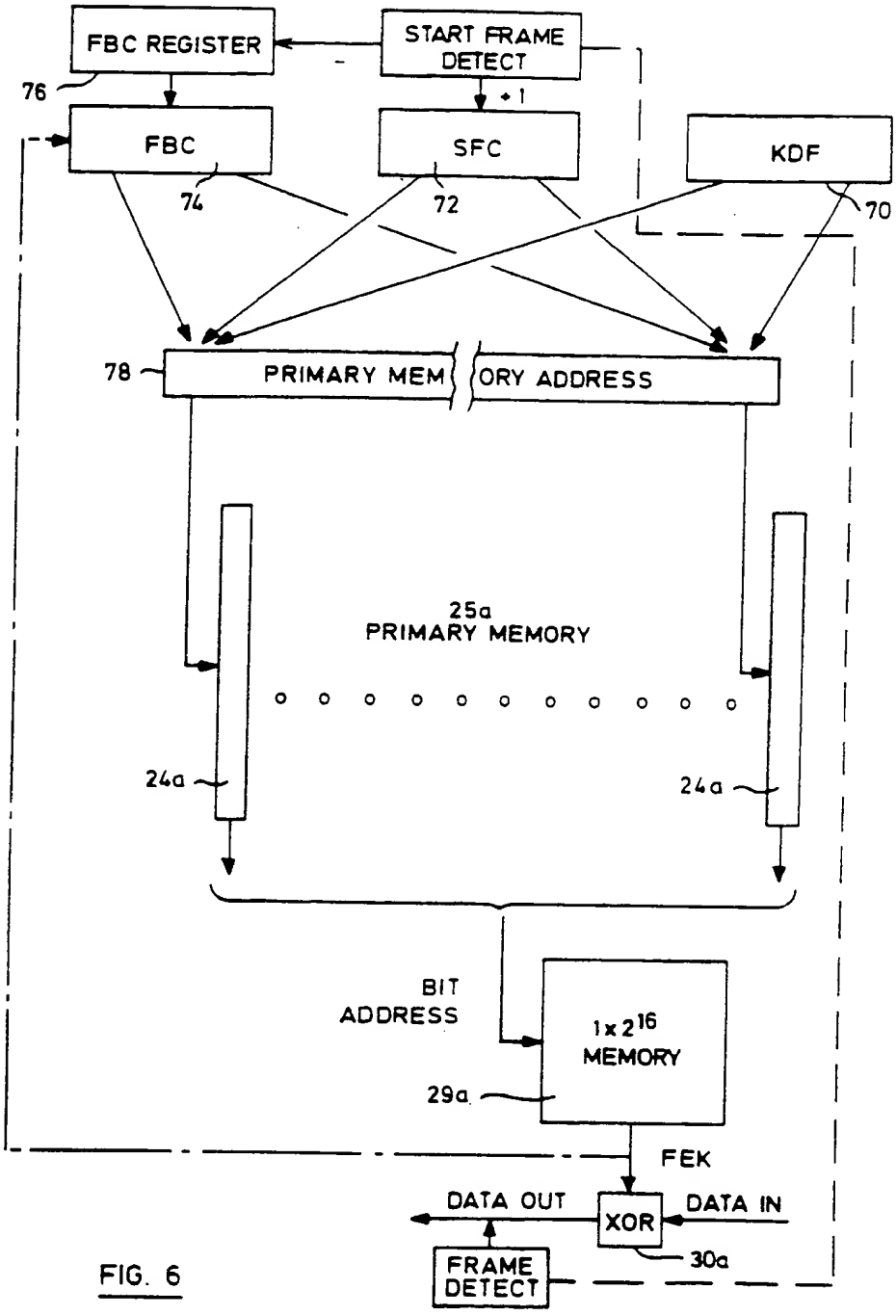


FIG. 6

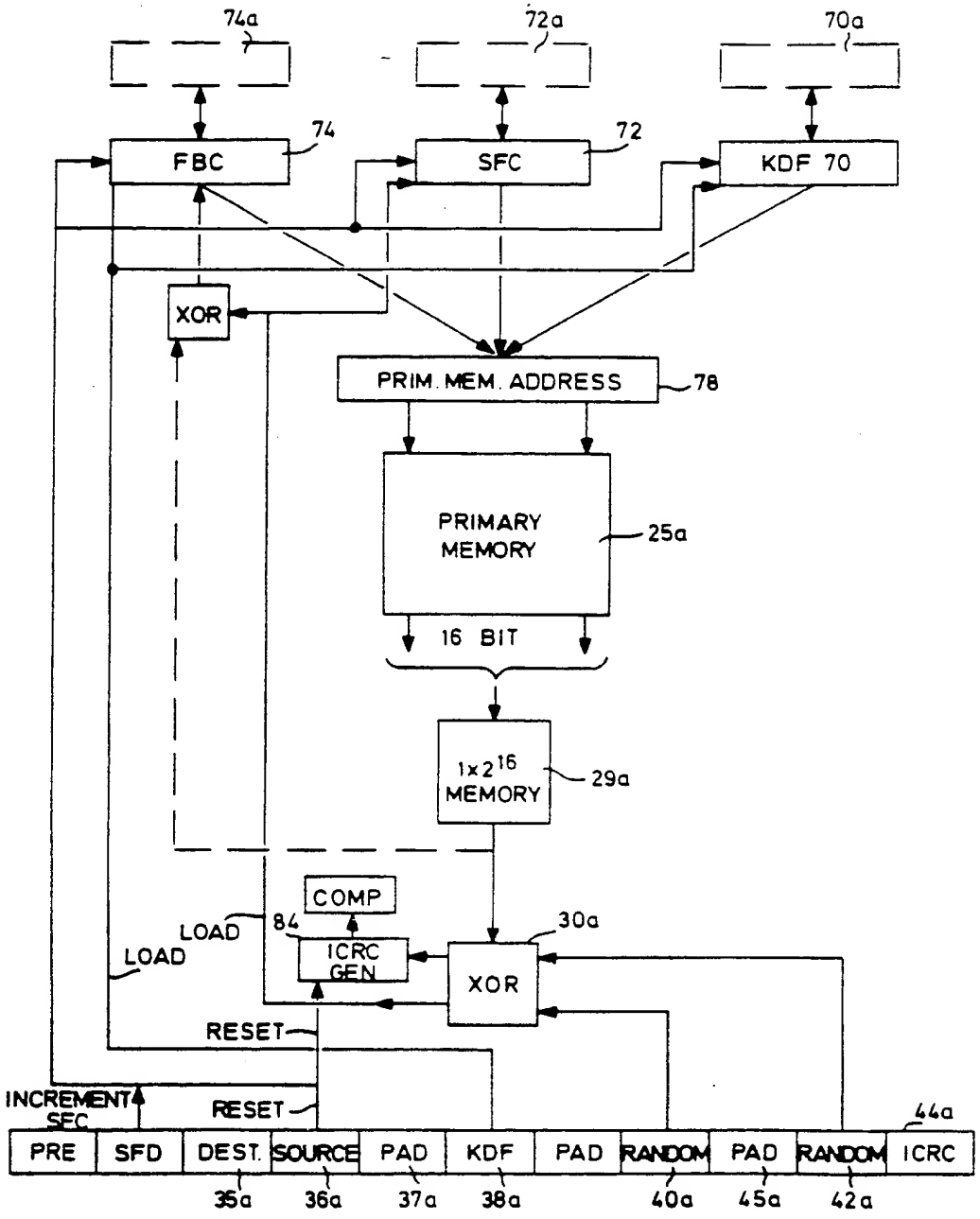


FIG 7

SUBSTITUTE SHEET

INTERNATIONAL SEARCH REPORT

International Application No PCT/CA 92/00486

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all)[6]

According to International Patent Classification (IPC) or to both National Classification and IPC

Int.Cl.5 H 04 L 29/06 H 04 L 9/08

II. FIELDS SEARCHED

Minimum Documentation Searched[7]

Classification System	Classification Symbols
Int.Cl.5	H 04 L

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched[8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT[9]

Category °	Citation of Document, [11] with indication, where appropriate, of the relevant passages [12]	Relevant to Claim No.[13]
A	COMPUTERS & SECURITY. INTERNATIONAL JOURNAL DEVOTED TO THE STUDY OF TECHNICAL AND FINANCIAL ASPECTS OF COMPUTER SECURITY vol. 8, no. 3, May 1989, AMSTERDAM NL pages 209 - 221 , XP71441 M.SHAIN 'SECURITY IN ELECTRONIC FUNDS TRANSFER' see paragraph 4.2 see paragraph 6.1 ---	1,15
A	IEEE COMMUNICATIONS MAGAZINE. vol. 23, no. 9, September 1985, US pages 41 - 46 D.M.BALENSON 'AUTOMATED DISTRIBUTION OF CRYPTOGRAPHIC KEYS USING THE FINANCIAL INSTITUTION KEY MANAGEMENT STANDARD' -----	1,15

° Special categories of cited documents : [10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
18-01-1993	15. 04. 93
International Searching Authority	**Signature of Authorized Officer**
EUROPEAN PATENT OFFICE	

Form PCT/ISA/210 (second sheet) (January 1985)